Can We Reconcile Robustness and Efficiency in Unsupervised Learning?

Moritz Hardt* Ankur Moitra†

November 21, 2012

Abstract

We consider a fundamental problem in unsupervised learning: given a collection of m points in $\mathbb{R}^n$, if many but not necessarily all of these points are contained in a d-dimensional subspace T can we find it? The points contained in T are called *inliers* and the remaining points are *outliers*. This problem has received considerable attention in computer science and in statistics. Yet efficient algorithms from computer science are not robust to *adversarial* outliers, and the estimators from robust statistics are hard to compute in high dimensions. This is a serious and persistent issue not just in this application, but for many other problems in unsupervised learning.

Are there algorithms for linear regression that are both robust to outliers and efficient? We give an algorithm that finds T when it contains more than a $\frac{d}{n}$ fraction of the points. Hence, for say $d = n/2$ this estimator is both easy to compute and well-behaved when there are a constant fraction of outliers. We prove that it is small set expansion hard to find T when the fraction of errors is any larger, thus giving evidence that our estimator is an *optimal* compromise between efficiency and robustness. In fact, this basic problem has a surprising number of connections to other areas including small set expansion, matroid theory and functional analysis that we make use of here.

*IBM Research Almaden. Email: mhardt@us.ibm.com

†Institute for Advanced Study and Princeton University. Email: moitra@ias.edu. Research supported in part by NSF grant No. DMS-0835373 and by an NSF Computing and Innovation Fellowship.

1 Introduction

Unsupervised learning refers to the problem of trying to find hidden structure in unlabeled data. A ubiquitous approach is to model this hidden structure as a low-dimensional subspace that contains many of the data points. This approach has found a range of applications in areas such as feature selection, dimensionality reduction, spectral clustering, topic modeling and statistical inference. There are two important desiderata for an unsupervised learning algorithm, *computational efficiency* and *robustness*: computational efficiency refers to the goal of giving provable guarantees on the running time of the algorithm and robustness refers to the goal of giving guarantees that the algorithm produces a useful output even if the assumptions of the model do not hold exactly. Our focus in this paper is on understanding whether or not these two goals can be met simultaneously.

Individually, these goals can each be met. For example, there are many known fast algorithms to compute the singular value decomposition, and from this decomposition it is straightforward to find a low-dimensional subspace that contains *all* of the data if it exists. There are also a number of provably robust estimators for linear regression. One famous example is Rousseeuw's *least median of squares* estimator [37]. The computational problem that underlies this estimator is to find a subspace that minimizes the median Euclidean distance to the data points. An adversary must corrupt at least half of the data points in order to corrupt the output. Many more robust estimators have been developed for this specific problem (e.g. least trimmed squares, M-estimators, the Theil-Sen estimator, reweighed least squares) and for other inference problems by the robust statistics community (see e.g. [38] and [24]).

Unfortunately, the singular value decomposition is not robust to outliers. Moreover, only modest improvements over brute-force search are known to actually compute the least median of squares estimator in high dimensions [17]. Is there an estimator for linear regression that is both efficiently computable and robust to outliers? This is an instance of a fundamental and largely unexplored question:

"Can we reconcile computational efficiency and robustness in unsupervised learning?"

Our focus here is on a challenging notion of robustness used in the robust statistics community: an estimator is robust if an adversary can corrupt an α fraction of the data, and the output of the estimator is still well-behaved. The fraction of data that an adversary is allowed to corrupt is called the *breakdown point* [15]. We remark that there has been interesting recent work on finding a subspace that approximately minimizes the sum of ℓ_p distances (for $p > 2$) to the data points [14], [23]. Unfortunately ℓ_p-regression can be corrupted quite easily by an adversary.

In general, the robust statistics community studies the breakdown properties of particular estimators. Here, our goal is not to study a particular estimator, but rather whether or not there is *any* robust estimator for linear regression that is also easy to compute. The following definition is central to our paper:

Definition 1.1. An estimator $\mathcal{E}$ is an *α-robust estimator* for the d-dimensional linear regression problem in $\mathbb{R}^n$ if for any set of points in which a $1-\alpha$ fraction are contained in a d-dimensional linear subspace $T \subseteq \mathbb{R}^n$, the estimator returns T.

Here the breakdown point is α. So the natural question is, for what choices of the parameters n, d and α is there such an estimator that is also easy to compute? There are compelling reasons

to choose robust estimators over their classical counterparts, but so far their potential has not been realized because there are no good algorithms to compute them.

1.1 Complexity of Robust Linear Regression

We assume that the points outside T are in general position, and that the points inside T are in general position with respect to T.[1] Recall that the dimension of T is d. Throughout this paper, we will use L to denote the points inside T and we will call these the *inliers*, and the remaining points *outliers*. Our first result is a simple randomized algorithm that achieves a breakdown point of $\alpha = 1 - \frac{d}{n}$. Our result relies on Condition 2.1: any set of n points is linearly independent if and only if at most d of the points are inliers.

Theorem 1.2. *If a set of m points in $\mathbb{R}^n$ has strictly more than $\frac{d}{n}m$ inliers and meets Condition 2.1, then there is a Las Vegas algorithm whose output is the set L of inliers, each iteration can be implemented in polynomial time and the expected number of iterations is $O(n^2 m)$.*

In fact, an interesting comparison can be drawn between our algorithm and the famous *RANSAC* method of Fischler and Bolles [19]: Both approaches repeatedly select a random set of n points; *RANSAC* works when this sample contains *only* inliers, whereas our algorithm works when the sample contains at least $d+1$ inliers. The main observation is that even if a set of n points contains many outliers, only the inliers can participate in a linear dependence. In fact, for $d = n/2$ and even if inliers make up 3/4 of the points, *RANSAC* will take an exponential number of iterations to find T while our algorithm requires only a constant number of iterations (see Remark 2.3).

Our algorithm can also be made stable in that the inliers do not need to be exactly contained within T. Here we need Condition 2.5: the smallest determinant of any set of points with at most d inliers is strictly larger than the largest determinant of any set of points with at least $d+1$ inliers.

Theorem 1.3. *If a set of m points in $\mathbb{R}^n$ has strictly more than $\frac{d}{n}m$ inliers and meets Condition 2.5, then there is a Las Vegas algorithm whose output is the set L of inliers, each iteration can be implemented in polynomial time and the expected number of iterations is $O(n^2 m)$.*

Our estimator achieves a constant breakdown point for, say, $d = n/2$. Yet there are numerous inefficient estimators that achieve a better breakdown point (e.g. a constant breakdown point even when $d = n-1$). We provide evidence that our estimator is the *optimal* compromise between efficiency and robustness: it is *small set expansion* hard to improve the breakdown point beyond this threshold. We state our result informally here:

Theorem 1.4. *There is an efficient reduction from an instance of (ε, δ)-Gap-Small-Set Expansion on a graph G to* Gap Inlier *such that:*

- *if there is a small non-expanding cut in G then there exists a subspace of dimension d containing at least $(1-\varepsilon)\frac{d}{n}$ fraction of the points*

[1] If we remove these conditions then when $dim(T) = n-1$ the problem is equivalent to trying to satisfy as many equations as possible in an overdetermined linear system. See [22], [28] and references therein. However, these reductions produce instances which are quite far from ones we might expect to observe in real data, and one approach for circumventing these hardness results is to instead require the above condition which is satisfied almost surely in most natural probabilistic models and seems to make the problem computationally much easier.

- *and if there is no small non-expanding cut then every subspace of dimension d contains at most a $2\varepsilon\frac{d}{n}$ fraction of the points.*

Khachiyan [27] proved a related result that it is NP-hard to find a $d = n-1$ dimensional subspace that contains a $(1-\varepsilon)\frac{n-1}{n}$ fraction of the data points[2]. In general, it seems difficult to base hardness for robust linear regression (when $d < n-1$) on standard assumptions and this is an interesting open question.

Taking a step back, computational complexity is an important lens for understanding learning and statistical problems in the sense that there are many sample-efficient estimators, e.g., maximum likelihood, that are hard to compute, but by allowing more samples than the information theoretic minimum we can find alternatives that are easy to compute. Yet these hard estimators are still favored in practice, perhaps not just due to their sample efficiency but also due to their robustness. A broader goal of our paper is to bring to light questions about whether there are estimators that meet all three objectives of being efficiently computable, sample efficient and robust (and not just two out of three).

1.2 Derandomization and Duality for Robust Linear Regression

The crucial step in our randomized algorithm is to repeatedly sample subsets of n points and once we find one that is linearly dependent, we can use this subset to recover the set of inliers. If a collection of m points in $\mathbb{R}^n$ has the property that a random subset of n points is linearly dependent (with non-negligible probability), can we find such a subset deterministically? We give a solution to this problem using tools from matroid theory [18], [11], [21]:

Indeed, a well-studied polytope in matroid literature is the *basis polytope* which is the convex hull of all sets of n points that form a basis (see Section 4). Condition 2.1 guarantees us that the vector $\frac{n}{m}\mathbf{1}$ is outside the basis polytope, and our goal of finding a set of n points that do not span $\mathbb{R}^n$ can be stated equivalently as finding a Boolean vector (whose coordinates sum to n) that is also outside the basis polytope.

There has been a vast literature on the basis polytope and on submodular minimization, and there are deterministic strongly polynomial time algorithms for deciding membership in the basis polytope [18], [11], [21], [39], [26]. Our idea is in each step we find a line segment ℓ that contains the current vector (starting with $\frac{n}{m}\mathbf{1}$). Since the current vector is outside the basis polytope it is easy to see that at least one of the endpoints of ℓ must also be outside. So we can move the current vector to this endpoint and if we choose these segments ℓ in an appropriate way we will quickly find a Boolean solution. The key is that a membership oracle for the basis polytope tells us which endpoint of ℓ we should move to. Hence we obtain an algorithm that is not only an optimal tradeoff between efficiency and robustness, but is even deterministic:

Theorem 1.5. *If a set of m points in $\mathbb{R}^n$ has strictly more than $\frac{d}{n}m$ inliers and meets Condition 2.1, then there is a deterministic polynomial time algorithm whose output is the set L of inliers.*

The basis polytope not only plays a central role in robust linear regression but is also closely related to a notion studied in functional analysis that we call *radial isotropic position*. In fact, Barthe [2] studied a convex programming problem whose optimal solution finds a linear transformation that places a set of points in radial isotropic position (see Section 6) if it exists.

[2]This follows by applying a padding argument to the knapsack instance before proceeding with the reduction in [27].

The connection is that the optimal value to this convex program is finite (i.e. there is such a transformation) if and only if the vector $\frac{n}{m}\mathbf{1}$ is inside the basis polytope.

Barthe's convex program provides a connection between radial isotropic position and robust linear regression: just as placing a set of points in isotropic position is a proof that the set of points is not contained in a low-dimensional subspace, so is placing a set of points in radial isotropic position a proof that there is no d-dimensional subspace that contains more than a $\frac{d}{n}$ fraction of the points (see Section 4). We give effective bounds on the region in which an optimal solution to the convex program is contained, and how strictly convex the function is and use this to give an efficient algorithm to compute radial isotropic position.

Theorem 1.6 (informal)**.** *There is a deterministic polynomial time algorithm to compute a linear transformation R that places a set of points in radial isotropic position, if such a transformation exists.*

Notably, this theorem shows that if there is no low-dimensional subspace that contains many of the points, we can deterministically compute a certificate that there is no such subspace.

Radial isotropic position can also be thought of as a more stable analogue of isotropic position that is not sensitive to either the norms of the data points or to a constant fraction of adversarial outliers! Isotropic position has important applications both in algorithms and in exploratory data analysis, but is quite sensitive to even a small number of outliers (see e.g. [40]). Just as robust statistics asks for estimators that are well-behaved in the presence of outliers, we could ask for *canonical forms* (e.g. isotropic position, radial isotropic position) that are well-behaved in the presence of outliers. Perhaps radial isotropic position will be a preferable alternative in some existing applications where being robust is crucial.

Somewhat surprisingly, this elementary problem of finding a low-dimensional subspace that contains many of the data points is connected to a number of problems and combinatorial objects including the small set expansion hypothesis, the independent set polytope and submodular minimization and notions in functional analysis, and we make use of all of these connections.

1.3 Related Work

Our work fits into a broader agenda within statistics and machine learning: Can we recover a low-rank matrix from noisy or incomplete observations? The foundational work of Recht, Fazel and Parrilo [36] and Candes and Recht [6] gave convex programming algorithms that provable recover a low-rank matrix when given a small number of random chosen entries in the matrix. These techniques have since been adapted to settings in which an adversary can corrupt some of the entries in the matrix [9], [5], [41]. However we note that there are two incomparable models for how an adversary is allowed to corrupt the entries in a low-rank matrix, and which model is more natural depends on the setting. For example, the exciting work of Candes et al [5] considers a model in which an adversary can corrupt a constant fraction of the entries of A whose locations are chosen uniformly at random. In contrast, the model in [41], [42] for example allows an adversary to corrupt a large fraction of the columns of A. This is the setting in our work, and this assumption is most natural when we think of columns of A as representing individuals from a population and uncorrupted columns correspond to individuals that fit the model, but we would like to make as few assumptions as possible about the remaining individuals that do not fit the model. We note that much of the recent work from statistics and machine learning has focused on stochastic settings for this problem, where one posits a distributional model that generates both the inliers and outliers and the goal is to recover the subspace T with high probability. Yet, our deterministic condition (Condition 2.1) is

usually satisfied in these probabilistic models. For example, the recent work of Lerman et al [30] considers a model in which inliers are chosen according to a standard Gaussian restricted to T, and outliers are chosen according to a standard Gaussian on $\mathbb{R}^n$. Samples chosen from this model satisfy Condition 2.1 with an exponentially small failure probability, and we believe that our randomized algorithm is a simpler and more attractive alternative to convex programming approaches for these problems.

The above discussion has focused on notions of robustness that allow an adversary to corrupt a constant fraction of the entries in the matrix A. However, this is only one possible definition of what it means for an estimator to be robust to noise. For example, principal component analysis can be seen as finding a d-dimensional subspace that minimizes the sum of squared distances to the data points. A number of works have proposed modifications to this objective function (along with approximation algorithms) in the hopes that this objective function is more robust. As an example, Deshpande et al [14] gave a $O(p^{p/2})$ approximation algorithm for the problem of finding a subspace that minimizes the sum of ℓ_p distances to the data points (for $p > 2$). Another example is the recent work of Naor, Regev and Vidick [32] which gives a constant factor approximation for finding a d-dimensional subspace that maximizes the sum of Euclidean lengths of the projections of the data points (instead of the sum of squared lengths). Lastly, we mention that Dunagan and Vempala [16] gave a geometric definition of an outlier (that does not depend on a hidden subspace T) and give an optimal algorithm for removing outliers according to this definition.

2 A Simple Randomized Algorithm

Here we give a randomized algorithm for robust linear regression. The idea is that once we find any non-trivially sparse linear dependence we can use it to find the set of inliers provided that the inliers are in general position with respect to T. The breakdown point of this estimator is exactly the threshold at which a random set of n points is linearly dependent with non-negligible probability. Surprisingly, in Section 3 we give evidence based on the small set expansion conjecture that there is no efficient estimator that has a better breakdown point.

We will think of an instance of robust linear regression as a matrix $A \in \mathbb{R}^{n \times m}$ with $m \geqslant n$ and rank n. Throughout this paper for $V \subset [m]$, we will let A_V denote the submatrix corresponding to columns in V. Suppose that there is a d-dimensional subspace T that contains *strictly* more than a $\frac{d}{n}$ fraction of the columns of A. Our goal is to recover this subspace (under mild general position conditions on these points) efficiently. Let $L \subset [m]$ be the columns of A that are inliers. We will need the following condition which is almost surely satisfied by any reasonable probabilistic model that generates inliers from the subspace T and outliers from all of $\mathbb{R}^n$:

Condition 2.1. *A set of n columns of A is linearly independent if and only if at most d of the columns are inliers.*

The next lemma gives a lower bound on the probability of sampling strictly more than expected number of inliers:

Lemma 2.2. *Suppose that we are given a set of m points in $\mathbb{R}^n$ with strictly more than $\frac{d}{n}m$ inliers. Let V be a uniformly random set of n points (without repetition). Then the probability that U contains at least $d+1$ inliers is at least $p \geqslant \frac{1}{2n^2 m}$.*

Algorithm 1 RandomizedFind, **Input:** $A \in \mathbb{R}^{n \times m}$ which satisfies Condition 2.1

1. Set $U = [m]$
2. *start* : Choose $V \subset U$ with $|V| = n$ uniformly at random
3. If $rank(A_V) < n$,
4. $\quad u \in ker(A_V)$, Set $\mathcal{L} = span(\{A_i : u_i \neq 0\})$, Set $L = \{i : A_i \in \mathcal{L}\}$
5. $\quad$ Output L
6. Else
7. $\quad$ Return to *start*

Proof: Let X be a random variable defined to be the number of inliers in a random set V of n points. Then $E[X] > d$ and set $\widehat{X} = X - E[X]$. Then let p be the probability that $\widehat{X} \geqslant 0$, and this condition certainly implies that we have at least $d+1$ inliers. Since the expectation of $\widehat{X}$ is zero, we have that

$$pE[\widehat{X}|X \geqslant 0] + (1-p)E[\widehat{X}|X < 0] = 0$$

Then we can upper bound $E[\widehat{X}|X \geqslant 0] \leqslant n - d$ and $-pE[\widehat{X}|X < 0] \leqslant pn$. Hence

$$p(n-d) + pn \geqslant -E[\widehat{X}|X < 0] \geqslant \frac{\lfloor \frac{d}{n} m \rfloor + 1}{m} - \frac{d}{n} \geqslant \frac{1}{nm}$$

and this completes the proof of the lemma. ■

Remark 2.3. *We remark that the lower bound on p can be improved to $p \geqslant (d/n)^2/2$ when $m \geqslant 6n+2$ and $n \geqslant 3$. Hence our algorithm is quite practical in this range of parameters.*

Indeed, with the same notation as above, condition X on the event E that the first two samples are contained in L (the set of inliers). Clearly, $\mathbb{P}\{E\} \geqslant (d/n)^2$. On the other hand, we still sample $n-2$ points with replacement. Each sample now has a probability of landing in L that is at least $q \geqslant \frac{dm/n-2}{m-2} = \frac{d}{n} - \frac{n+d}{n(m-2)} \geqslant \frac{d}{n} - \frac{1}{3n}$. Here, we used that $m \geqslant 6n+2$. Hence, $\mathbb{E}[X \mid E] \geqslant (n-2)(\frac{d}{n} - \frac{1}{3n}) \geqslant d - 1$, where we used that $2/n + 1/3 \leqslant 1$. On the other hand, we have $\mathbb{P}\{X \geqslant \lfloor \mathbb{E}[X \mid E] \rfloor \mid E\} \geqslant \frac{1}{2}$ by the "mean is median" theorem for hypergeometric distributions (see, e.g., [25]). It follows that $\mathbb{P}\{X \geqslant d+1\} \geqslant \mathbb{P}\{X \geqslant d-1 \mid E\}\mathbb{P}\{E\} \geqslant \frac{1}{2}(d/n)^2$.

The next claim captures the intuition that from any non-trivially sparse linear dependence in A it is easy to compute the set of inliers:

Claim 2.4. *If $|V| = n$, then any vector in the kernel of A_V must contain $d+1$ inliers in its support and no outliers.*

Proof: Suppose there is a vector $u \in ker(A_V)$ with $u_i \neq 0$ for $i \notin L$. Since any $d+1$ inliers are linearly dependent, we can use Caratheodory's Theorem (see [31]) to find a vector $v \in ker(A_V)$ supported on at most d inliers and for which $v_i \neq 0$. This contradicts Condition 2.1 since the support of any non-zero vector in the kernel must contain at least $d+1$ outliers (otherwise we can extend the support to a basis). Furthermore, any vector in the kernel of A_V must be supported on at least $d+1$. ■

Algorithm 2 RandomizedFind2, **Input:** $A \in \mathbb{R}^{n \times m}$ which satisfies Condition 2.5

1. Set $U = [m]$
2. *start* : Choose $V \subset U$ with $|V| = n$ uniformly at random
3. If $det(A_V^T A_V) < C^2$
4. While $|V| > d + 1$
5. Find $\{u\}$ such that $det(A_{V-\{u\}}^T A_{V-\{u\}}) < C^2$
6. Set $V = V - \{u\}$
7. Set $\mathcal{L} = V \cup \{v | det(A_{V\Delta\{u,v\}}^T A_{V\Delta\{u,v\}}) < C^2\}$ where $u \in V$
8. Else
9. Return to *start*

We can now prove Theorem 1.2:

Proof: Claim 2.4 guarantees the correctness of the algorithm, and Lemma 2.2 guarantees that the success probability of each iteration is at least $p \geqslant \frac{1}{2n^2 m}$ and this implies the lemma. ■

We are interested in generalizing our algorithm to the setting where inliers are only approximately contained in the subspace. This idea is formalized next.

Condition 2.5. *Any set V of at most n columns of A has $det(A_V^T A_V) \geqslant C^2$ if the number of inliers is at most d, and otherwise strictly less than C^2.*

We can now prove Theorem 1.3, which is stable even when the inliers are not exactly contained in a subspace T:

Proof: Lemma 2.2 guarantees that the probability that the algorithm finds a set V with $|V| = n$ and $det(A_V) < C$ in is at least $p \geqslant \frac{1}{2n^2 m}$, and furthermore the algorithm maintains the invariant that the set V always has at least $d + 1$ inliers, and at the end of the while loop V is a set of $d + 1$ inliers. Then Condition 2.5 guarantees that the algorithm correctly outputs the set of inliers. ■

3 Computational Limits

We will now present evidence that the robust linear regression problem is computationally hard beyond the breakdown point achieved by our randomized algorithm in Section 2. For this purpose we need to introduce the *expansion profile* of a graph. Given a Δ-regular graph $G = (V, E)$ we define the edge expansion of a set $S \subseteq V$, as

$$\phi_G(S) = \frac{|E_G(S, V \backslash S)|}{\Delta |S|}.$$

Here and in the following, we let $E_G(A, B)$ denote the set of edges in G with one endpoint in A and the other in B. Let us also denote $\mu(S) = |S|/|V|$. Given a parameter $\delta \in [0, 1/2]$, we define the *expansion profile* of G as the curve

$$\phi_G(\delta) = \min_{\mu(S)=\delta} \phi(S).$$

With these definitions we describe the Small Set Expansion problem as was recently studied by [34, 35]:

Definition 3.1. The Gap-Small-Set Expansion problem is defined as: Given a graph G, and constants $\varepsilon, \delta > 0$, distinguish the two cases

1. $\phi_G(\delta) \geqslant 1 - \varepsilon$,
2. $\phi_G(\delta) \leqslant \varepsilon$.

We will relate the previous problem to the Gap-Inlier problem that we define next.

Definition 3.2. The Gap-Inlier problem is defined as: Given m points $u_1, \ldots, u_m \in \mathbb{R}^n$, and constants ε, δ, distinguish the two cases

1. there exists a subspace of dimension δn containing a $(1-\varepsilon)\delta$ fraction of the points,
2. every subspace of dimension δn contains at most a $\varepsilon\delta$ fraction of the points.

Our next theorem shows a reduction from Gap-Small-Set Expansion to Gap Inlier.

Theorem 3.3. *Let $\varepsilon, \delta > 0$. There is an efficient reduction which given a Δ-regular graph $G = (V, E)$, produces an instance $u_1, \ldots, u_m \in \mathbb{R}^n$ of* Gap-Inlier *such that*

Completeness: *If $\phi_G(\delta) \leqslant \varepsilon$, then there exists a subspace of dimension δn containing at least $(1-\varepsilon)\delta$ fraction of the points.*

Soundness: *If $\phi_G(\delta') \geqslant 1 - \varepsilon$ for every $\delta' \in [2\delta/\Delta, 2\delta]$, then every subspace of dimension δn contains at most a $2\varepsilon\delta$ fraction of the points.*

Proof: Our reduction works as follows. Let $G = (V, E)$ be an instance of Gap-Small-Set Expansion. Let $m = |E|$ and $n = |V|$. For each edge $e = (i, j)$, create a vector $u_e = \alpha_e e_i + \beta_e e_j$, where e_i is the i-th standard basis vector and α_e, β_e are drawn independently and uniformly at random from $[0, 1]$. This defines an instance $u_1, \ldots, u_m \in \mathbb{R}^n$ of Gap-Inliers.

To analyze our reduction, it will be helpful to consider the following intermediate graph. Let $B = (E, V)$ be the bipartite graph where we connect each edge $e \in E$ with the two vertices in V that it is incident to. Note that B is $(2, \Delta)$-regular. The next claim relates the dimension of a set of points to the size of the neighborhood of the corresponding edge set in the graph B.

Claim 3.4. *For every set of points $P \subseteq \{u_1, \ldots, u_m\}$ corresponding to a set of edges $F \subseteq E$ we have with probability 1 over the choice of the coefficients above*

$$\dim(\mathrm{span}(P)) = |E_B(F, V)| .$$

Proof: On the one hand, the points P are contained in the coordinate subspace of dimension $d = |E_B(F, V)|$ corresponding to the union of the support of the vectors. On the other hand, we claim that that they also span this coordinate subspace. Fix any set of d points touching all d coordinates. It is not difficult to show that these points are linearly independent with probability 1 over the randomness in the coefficients.[3] ■

[3] Note that without perturbation an even cycle, for example, causes a linear dependence.

Completeness. We begin with the completeness claim. Let $S \subseteq V$ be a set of measure δ and suppose that $\phi_G(S) \leqslant \varepsilon$. Double-counting the edges spanned by S, we get

$$\Delta|S| = 2|E_G(S,S)| + |E_G(S, V\setminus S)|.$$

Hence, $|E(S,S)| \geqslant \Delta|S|/2 - \varepsilon\Delta|S|/2 = (1-\varepsilon)\Delta|S|/2$. On the other hand the edge set $E_G(S,S)$ has at most $|S|$ neighbors in B. This implies that the points corresponding to $E(S,S)$ are contained in a coordinate subspace of dimension $|S|$. Equivalently, there exists a (δn)-dimensional subspace containing at least $(1-\varepsilon)\delta\Delta n/2$ points. Since $m = \Delta n/2$, this corresponds to a fraction of $(1-\varepsilon)\delta$ which is what we wanted to show.

Soundness. Next we establish soundness. Consider any set of points P contained in δn dimensions. We will show that under the given assumption on the expansion profile of G, it follows that $|P| \leqslant \varepsilon\Delta\delta n$. Again, since $m = \Delta n/2$, this directly implies that any subspace of dimension δn contains at most a $2\varepsilon\delta$ fraction of the points. Let F be the set of edges corresponding to P in the graph B and let S be its vertex neighborhood in B. By Claim 3.4, it is sufficient to show that $|F| \leqslant \varepsilon\Delta\delta n$. First, note that the neighbor set $S \subseteq V$ of F in the graph B satisfies

$$\frac{\delta n}{2\Delta} \leqslant |S| \leqslant 2\delta n. \tag{1}$$

The second inequality follows from the fact that each edge e has exactly two neighbors and we have equality if the edges form a matching. The first inequality follows because G is Δ-regular. Thus, a set of $|S|$ vertices can induce at most $\Delta|S|/2$ edges and all edges in F are induced by S.

Counting the edges touching S as before,

$$\Delta|S| = 2|E_G(S,S)| + |E_G(S, V\setminus S)| \geqslant 2|E_G(S,S)| + \Delta(1-\varepsilon)|S|.$$

The inequality followed from our assumption on the expansion profile of G which we may apply because S satisfies Equation 1. Consequently:

$$|E_G(S,S)| \leqslant \frac{\varepsilon\Delta|S|}{2}.$$

On the other hand, $|F| \leqslant |E_G(S,S)|$, since every edge in F is induced by S. Hence, the previous inequality showed that $\varepsilon\Delta\delta n/2 \geqslant |F|$. ■

4 The Basis Polytope

Here we connect the *independent set polytope* which has received considerable attention in matroid literature, to a notion studied in functional analysis that we call *radial isotropic position*. In Section 5 we will use known algorithms for deciding membership in the independent set polytope to derandomize our algorithm from Section 2. And in Section 7 we will give an efficient algorithm to compute radial isotropic position, which can be thought of as a robust analogue to isotropic position. Let $A = [u_1, \ldots, u_m] \in \mathbb{R}^{n\times m}$ with $m \geqslant n$.

Definition 4.1. Let P be the *independent set polytope* defined as:

$$P \stackrel{\text{def}}{=} \operatorname{conv}\left\{\mathbf{1}_U \colon U \subseteq [m], \dim\left(\operatorname{span}\{u_i \colon i \in U\}\right) = |U|\right\},$$

where $\mathbf{1}_U$ is the m-dimensional indicator vector of the set U. Also let K_A be the *basis polytope* which is the facet of P corresponding to $\sum_i x_i = n$.

These polytopes can be defined (in a more general context) using the language of matroid theory where independent sets of vectors are replaced by independent sets in a matroid. A fundamental algorithmic problem in matroid theory is to give an efficient membership oracle for these polytopes. A number of solutions are known which all follow from a characterization of Edmonds [18] that reduces membership to solving a submodular minimization problem:

$$\min_{U \subset [m]} \operatorname{rank}(\{u_i : i \in U\}) - \sum_{i \in U} x_i$$

The optimum value of this minimization is nonnegative if and only if $x \in P$ [18]. Hence an immediate consequence of the known algorithms for submodular minimization [21], [39], [26] and even a direct algorithm of Cunningham [11] yield:

Theorem 4.2. *There is a deterministic polynomial time algorithm to solve the membership problem for the independent set polytope P (and the basis polytope K_A).*

We will use this tool from matroid theory to derandomize our algorithm from Section 2. Recall that the main step in our algorithm is to repeatedly sample subsets of n points and once we find one that is linearly dependent, we can use this subset to recover the set of inliers. So our approach is to use a membership oracle for the basis polytope to find a subset of n points that is linearly dependent deterministically.

The basis polytope not only plays a central role in robust linear regression but also in a notion studied in functional analysis called radial isotropic position. These two concepts can be thought of as dual to each other: Recall that the set of vectors $u_1, \ldots, u_m \in \mathbb{R}^n$ is in *isotropic position* if

$$\sum_{j=1}^{m} u_j \otimes u_j = \mathrm{Id}_n$$

It is well-known that a set of points can be placed in isotropic position if and only if the points are not all contained in an $n-1$-dimensional subspace. Just as isotropic position can be thought of as a certificate that a set of points is full-dimensional, so too radial isotropic position can be thought of as a certificate that there is no low-dimensional subspace that contains many of the points.

Definition 4.3. We say that a linear transformation $R\colon \mathbb{R}^n \to \mathbb{R}^n$ puts set of vectors $u_1, \ldots, u_m \in \mathbb{R}^n$ in *radial isotropic position* with respect to a coefficient vector $c \in \mathbb{R}^m$ if

$$\sum_{i=1}^{m} c_i \frac{Ru_i}{\|Ru_i\|} \otimes \frac{Ru_i}{\|Ru_i\|} = \mathrm{Id}_n$$

If a set of vectors meets Condition 2.1 then it cannot be put in radial isotropic position: any linear transformation A preserves the invariant that the inliers lie in a subspace of dimension d, but after applying A and rescaling the points to be unit vectors the variance of a random sample restricted to this subspace is strictly larger than d, which is too large! More generally, when can a set of vectors be put in radial isotropic position? Barthe [2] gave a complete answer to this question:

Theorem 4.4 (Barthe)**.** *A set of vectors $u_1, \ldots, u_m \in \mathbb{R}^n$ can be put in* radial isotropic position *with respect to $c \in \mathbb{R}^m$ if and only if $c \in K_A$. Moreover, $c \in K_A$ if and only if the following supremum has*

finite value:

$$\sup_{t_1,\ldots,t_m\in\mathbb{R}} \langle c,t\rangle - \log\det\left(\sum_{i=1}^{m} e^{t_i} u_i\otimes u_i\right)$$

This concave maximization problem provides a connection between radial isotropic position and robust linear regression. The optimal value reveals to us which case we are in: if it is finite, then the points can be put in radial isotropic position but if it is infinite (under Condition 2.1) then there is a subspace T of dimension d that contains more than a $\frac{d}{n}$ fraction of the points!

5 A Deterministic Algorithm

In this section we apply tools from matroid theory (see [18], [11]) to derandomize our algorithm from Section 2. Recall that the main step in our algorithm from Section 2 is to repeatedly sample subsets of n points and once we find one that is linearly dependent, we can use this subset to recover the set of inliers. Our goal is to find such a subset deterministically, and we can think about this problem instead in terms of the basis polytope.

Condition 2.1 guarantees that the vector $\frac{n}{m}\mathbf{1}$ is outside the basis polytope. We remark that a set of n columns is linearly dependent if and only if the indicator vector is outside the basis polytope. So we can think about this derandomization problem instead as a rounding problem: we are given a vector $\frac{n}{m}\mathbf{1}$ that is outside the basis polytope and we would like to round it to a Boolean vector (that sums to n) that is also outside the basis polytope.

Our approach is simple to describe, and builds on known polynomial time membership oracles for the basis polytope developed within combinatorial optimization [18], [11], [21], [39], [26]. In each step we find a line segment ℓ that contains the current vector (starting with $\frac{n}{m}\mathbf{1}$). Since the current vector is outside the basis polytope it is easy to see that at least one of the endpoints of ℓ must also be outside. So we can move the current vector to this endpoint and if we choose these segments ℓ in an appropriate way we will quickly find a Boolean solution.

Indeed, Edmonds gave a general characterization of the independent set polytope:

Theorem 5.1. *[18] The independent set polytope P can equivalently be described as:*

$$P \stackrel{\text{def}}{=} \left\{x\in\mathbb{R}^m\colon \text{ for all } U\subset[m], \dim\left(\operatorname{span}\{u_i\colon i\in U\}\right)\geqslant \sum_{i\in U} x_i\right\}$$

Hence we can intersect this alternative description of P with the constraint $\sum_i x_i = n$ to obtain an alternative description of the basis polytope that will be more convenient for our purposes. Indeed, if Condition 2.1 is met then any subset U of points has rank equal to $\min(n, \min(|U\cap L|, d) + |U/L|)$ and so:

Corollary 5.2. *If a set of $m\geqslant n$ points meets Condition 2.1, then*

$$P = \left\{x\in\mathbb{R}^m\colon 0\leqslant x_i\leqslant 1, \sum_{i=1}^{m} x_i\leqslant n \text{ and } \sum_{i\in L} x_i\leqslant d\right\}$$

$$K_A = \left\{x\in\mathbb{R}^m\colon 0\leqslant x_i\leqslant 1, \sum_{i=1}^{m} x_i = n \text{ and } \sum_{i\in L} x_i\leqslant d\right\}$$

Algorithm 3 DerandomizedFind, **Input:** $A \in \mathbb{R}^{n \times m}$ which satisfies Condition 2.1

1. Set $U = [m]$
2. While $|U| > n$
3. For each $i \in U$
4. Check if $\frac{n}{|U \setminus \{i\}|}\mathbf{1} \in K_{A_{U \setminus \{i\}}}$
5. If 'NO', Set $U = U \setminus \{i\}$ (exit for loop)
6. Find $u \in ker(A_U)$, Set $T = span(\{A_i : u_i \neq 0\})$, Return $L = \{i : A_i \in T\}$

Lemma 5.3. *After exiting the while loop,* $|U \cap L| \geqslant d + 1$

Proof: An immediate consequence of Corollary 5.2 is that for each call to the membership oracle for K_V for some set V, the answer is 'NO' if and only if the fraction of inliers in V is more than $\frac{d}{n}$. Since at the start of the while loop we are guaranteed that the fraction of inliers in U is more than $\frac{d}{n}$, this is an invariant of the algorithm. All that remains is to check that for any set U with $|U| > n$ and more than a $\frac{d}{n}$ fraction of inliers, there some element i that we can remove from U to maintain this condition (i.e. the algorithm does not get stuck). This is easy to check since if U contains even just one outlier, we can choose i to be that element and this will only increase the fraction of inliers and if instead there are no outliers left then we can choose any inlier to remove. Hence the algorithm does not get stuck, outputs a set U with $|U| = n$ which has strictly more than a $\frac{d}{n}$ fraction of inliers and so $|U \cap L| \geqslant d + 1$. ■

Theorem 5.4. *Given a set of m points $u_1, \ldots, u_m \in \mathbb{R}^n$ with $m \geqslant n$ that meets Condition 2.1 and which T contains more than a $\frac{d}{n}$ fraction of the points, then* DerandomizedFind *computes T. The running time of this algorithm is bounded by a fixed polynomial in n, m.*

Proof: Since $|U \cup L| \geqslant d+1$, we have that $rank(A_U) < n$. Then using Claim 2.4, DerandomizedFind computes the span T of the inliers, and outputs exactly the set of inliers. Note that there are a number of known strongly polynomial time algorithms for deciding membership in K_A (see Section 4). ■

6 Barthe's Convex Program

Recall that the basis polytope K_A characterizes exactly when we can put a set of points in radial isotropic position [2]. There are several known algorithms from the matroid literature that provide a strongly polynomial time algorithm for deciding membership in K_A. However, the focus of this section and the next is not just deciding if the optimization problem of Barthe has finite or infinite value, but finding an optimal solution in case that the optimum is finite. From the solution to this optimization problem, we will be able to derive the linear transformation that places a set of points in radial isotropic position. Here we will explain in detail the connection found by Barthe [2] and others [8, 7] between convex programming and radial isotropic position. In the next section we will prove various effective bounds on this convex programming problem that we need in order to show that the Ellipsoid method finds an optimal solution.

Recall that Barthe considers maximizing a concave function (or equivalently minimizing a convex function):

$$\sup_{t_1,\ldots,t_m\in\mathbb{R}} \langle c,t\rangle - \log\det\left(\sum_{i=1}^m e^{t_i} u_i\otimes u_i\right)$$

for a given set of points $u_1,\ldots,u_m\in\mathbb{R}^n$ and a coefficient vector $c\in\mathbb{R}^m$. How is this unconstrained maximization problem related to the linear transformation that puts the points $u_1,\ldots,u_m$ into radial isotropic position? For now we specialize our discussion to the case in which $c=\frac{n}{m}\mathbf{1}$, where $\mathbf{1}$ is the all ones vector. Let $t_1,\ldots,t_m\in\mathbb{R}$. Consider the matrix $U=\sum_{j=1}^m e^{t_j}u_j\otimes u_j$. We know that this matrix is positive definite and has full rank. Therefore it has a symmetric positive definite square root and we can define $R=U^{-1/2}$. Notice that

$$\mathrm{Id}_n = U^{-1/2}UU^{-1/2} = \sum_{j=1}^m e^{t_j} Ru_j\otimes Ru_j$$

Hence, we have what we need if we can choose t_j such that $e^{t_j}=\frac{n}{m}\|Ru_j\|^{-2}$. The crucial insight is that these conditions are exactly the optimality conditions in Barthe's maximization problem.

Lemma 6.1 ([2])**.** *Let $A=[u_1,\ldots,u_m]$ denote a matrix with column vectors $u_1,\ldots,u_m\in\mathbb{R}^n$. Suppose $\phi_A^*(c)<\infty$. Then, any optimal solution $t_1,\ldots,t_m$ to $\phi_A^*(c)$ satisfies $c_j=\langle e^{t_j}u_j,(Ae^TA^*)^{-1}u_j\rangle$ for every $1\leqslant j\leqslant m$.*

For completeness, we present Barthe's proof and to simplify notation we will continue specializing our discussion to $c=\frac{n}{m}\mathbf{1}$. Consider maximizing the function f over $\mathbb{R}^m$ defined as:

$$f(t)=\frac{n}{m}\sum_{j=1}^m t_j - \log\det U$$

It is not hard to show that f is concave (a short proof is given in Lemma 7.4). What is crucial is that if t maximizes $f(t)$, then it must satisfy that the gradient of f at t vanishes. We can apply a well-known formula for the derivative of $\log\det$ (see e.g. [29]) and:

$$0=\frac{\partial f(t)}{\partial t_j}=\frac{n}{m}-\mathrm{Tr}\left(U^{-1}\frac{\partial U}{\partial t_j}\right).$$

Also in our case:

$$\frac{\partial Ae^TA^*}{\partial t_j}=\sum_{j=1}^m\frac{\partial e^{t_j}u_j\otimes u_j}{\partial t_j}=e^{t_j}u_j\otimes u_j.$$

And so we conclude that the optimality condition is for all $j\in[m]$:

$$0=\frac{n}{m}-\mathrm{Tr}\left(U^{-1}e^{t_j}u_j\otimes u_j\right)=\frac{n}{m}-e^{t_j}\langle u_j,U^{-1}u_j\rangle.$$

where the last step uses the identity $\mathrm{Tr}(ABC)=\mathrm{Tr}(BCA)$. Recall that $\langle u_j,U^{-1}u_j\rangle=\|Ru_j\|^2$ and so any optimal $t\in\mathbb{R}^m$ satisfies $e^{t_j}=\frac{n}{m}\|Ru_j\|^{-2}$ which is precisely the condition we needed. And by the concavity of f, the supremum of $f(t)$ is attained if it is finite.

7 Computing Radial Isotropic Position Efficiently

Here we prove two important properties of the convex programming problem considered by Barthe, that we will need in order to prove that the Ellipsoid method can solve it. We prove that if the optimum is finite, there is a solution in a bounded region that is optimal. Also we establish a lower bound on how strictly convex the objective function is, since we will need this to show that any candidate solution that is close enough to achieving the optimum value must also be close to the optimum solution.

7.1 Effective Bounds

Here we prove bounds on the region in which an optimal solution can be found. Our proof follows the same basic outline as in [3, 2, 8] but is self-contained. We define $\phi_A \colon \mathbb{R}^m \to \mathbb{R}$ as

$$\phi_A(t_1,\ldots,t_m) = \log\det\left(\sum_{j=1}^{m} e^{t_j} u_j \otimes u_j\right)$$

Given $c \in \mathbb{R}^m$, consider the optimization problem $\phi_A^*(c) = \sup_{t\in\mathbb{R}^m}\langle t,c\rangle - \phi_A(t_1,\ldots,t_m)$. The function ϕ_A^* is the Legendre transform of ϕ_A. For convenience, we will write $\log\det(\sum_{j=1}^m e^{t_j} u_j \otimes u_j) = \log\det(Ae^T A^*)$ where T denotes the diagonal matrix with entries $t_1,\ldots,t_m$ and A^* is the transpose of A and e^T denotes the matrix exponential of T (i.e. a diagonal matrix in which the i^{th} entry on the diagonal is e^{t_i}). We also introduce the notation:

Definition 7.1. Let $d_I = \det(A_I A_I^*)$, $t_I = e^{\sum_{j\in I} t_j}$ and $D = \min_{I\colon d_I\neq 0} d_I$, where A_I is the sub matrix whose columns are indexed by I.

When we use the subscript I without further specification, we will always mean a subset of $[m]$ of size n. We will make repeated use of the Cauchy-Binet formula in this section:

Fact 7.2. *Let $A, B^* \in \mathbb{R}^{m\times n}$. Then* $\det(AB) = \sum_{I\colon |I|=n} \det(A_I)\det(B_I^*)$.

This generalizes the well-known identity that the determinant of the product of two matrices is the product of the determinants. We can apply this formula:

Claim 7.3. $\det\left(\sum_{j=1}^m e^{t_j} u_j \otimes u_j\right) = \det(Ae^T A^*) = \sum_{I\subseteq[m],|I|=n} t_I d_I$

We can now show that the mapping ϕ_A is convex, and hence $\phi_A^*(c)$ is concave (which we asserted in Section 6):

Lemma 7.4. *The function ϕ_A is convex on $\mathbb{R}^m$.*

Proof: Let $s,t \in \mathbb{R}^m$. Then, applying the Chauchy-Schwarz inequality,

$$\begin{aligned}\phi\left(\frac{s+t}{2}\right) &= \log\det(Ae^{(S+T)/2}A^*) = \log\left(\sum \sqrt{s_I d_I}\sqrt{t_I d_I}\right)\\ &\leqslant \log\left(\sqrt{\sum s_I d_I}\sqrt{\sum t_I d_I}\right) = \log\sqrt{\det(Ae^S A^*)} + \log\sqrt{\det(Ae^T A^*)} = \frac{\phi(s)+\phi(t)}{2}\end{aligned}$$

where the second equality uses Claim 7.3. ■

Our main step is to show that if c is contained in K_A with "sufficient slack", then the optimum is finite and we get a bound on the norm of an optimizer. To state the condition we need, we will use a slightly unconventional definition for how to dilate K_A. This simplifies our arguments (in part because it preserves the "trivial" constraints that the coordinates sum to n and are each in the interval $[0,1]$):

Definition 7.5. Let CK_A denote the vectors c whose coordinates sum to n and are each in the interval $[0,1]$ and for all nonnegative directions u with $u_{min} = 0$, $C \max_{v \in K_A} \langle u, v \rangle \geqslant \langle u, c \rangle$.

Lemma 7.6. *Let $\alpha > 0$ and suppose $c \in (1-\alpha)K_A$. Then*

1. $\phi_A^*(c) < \log \frac{1}{D}$,
2. *t^* with $f(t^*) = \phi_A^*(c)$ satisfies $\|t^*\|_\infty \leqslant \frac{2}{\alpha} \log \frac{1}{D}$*

Proof: From the assumption that $c \in K_A$, it follows directly that $\sum_{i=1}^m c_i = n$ and that $c_i \in [0,1]$. Throughout this proof, let $f(t) = \langle c, t \rangle - \phi_A(t)$. Let $t \in \mathbb{R}^m$. We need to upper bound $f(t)$. Note that we may assume that $\min_j t_j = 0$ by adding a constant $a \in \mathbb{R}$ to all coordinates without changing the value of $f(t)$. For notational convenience, assume that the coordinates of t are sorted in decreasing order $t_1 \geqslant t_2 \geqslant \ldots \geqslant t_m = 0$. This is without loss of generality since we can always apply a permutation to the columns of A and the coordinates of t without changing the function value.

Claim 7.7. $f(t) \leqslant \log\left(\frac{1}{D}\right) - \alpha \max_{j=1}^m t_j$

Proof: Let $I^* \subseteq [m]$ be the set of the n pivotal vectors in $[u_1, \ldots, u_m]$, i.e., the indices of the vectors that are not in the span of the vectors to the left of them.

By the monotonicity of the logarithm and Claim 7.3:

$$\phi_A(t_1, \ldots, t_m) = \log\Big(\sum t_I d_I\Big) \geqslant \log(t_{I^*} d_{I^*}) = \sum_{j \in I^*} t_j + \log(d_{I^*}) \geqslant \sum_{j \in I^*} t_j + \log\Big(\min d_I\Big).$$

Furthermore, we claim that

$$\sum_{j \in I^*} t_j - \sum_{j=1}^m c_j t_j \geqslant \alpha \max_j t_j. \tag{2}$$

Together these two inequalities directly imply that the statement of the claim. It therefore only remains to prove (2). First note that I^* maximizes $\langle \mathbf{1}_I, t \rangle = \sum_{j \in I} t_j$ among all I such that $d_I \neq 0$. On the other hand, we know that $c \in (1-\alpha)K_A$. Hence $\langle c, t \rangle \leqslant (1-\alpha)\langle \mathbf{1}_{I^*}, t \rangle$ and this implies

$$\sum_{j \in I^*} t_j - \sum_{j=1}^m c_j t_j \geqslant \alpha \sum_{j \in I^*} t_j \geqslant \alpha t_1$$

which establishes (2). ■

The previous claim shows that as any t_j tends to infinity, $f(t)$ tends to zero. Hence $\phi_A^*(c) < \infty$ and, by the convexity of ϕ_A, the supremum is attained meaning that we can find t^* such that $f(t^*) = \phi_A^*(c)$. But $f(t^*) = \phi_A^*(c) \geqslant f(0) = \log\det(AA^*) = \log\Big(\sum_I d_I\Big) \geqslant \log(\min_I d_I)$ where we used Claim 7.3 in the second inequality. Combining this inequality with Claim 7.7, we conclude

$$\max_j t_j^* \leqslant \frac{2}{\alpha} \log\left(\frac{1}{\min_I d_I}\right).$$

■

7.2 Strict Convexity

Here we prove that if a candidate solution t is close to achieving the optimal value then it is also close to the optimal solution t^*. This is not a vacuous property since if a convex function f is not strictly convex, being close to the optimal value for the objective function does not imply that a solution is close to the optimal solution.

The catch is that our function f is not strictly convex on all of $\mathbb{R}^m$. If t_a denotes the vector obtained from t by adding the constant a to all coordinates in t, then for every a, $f(t_a) = f(t)$ (where here we use the condition that $\sum_j c_j = n$). Hence, there are points t, t' at arbitrary distance that satisfy $f(t) = f(t')$. However, we can show that this is the only scenario in which the function is not strictly convex.

Definition 7.8. Let us say that $s, t \in \mathbb{R}^m$ are b-separated if $\|(s + a\mathbf{1}) - t\|_\infty \geqslant b$ for every $a \in \mathbb{R}$. Here, $\mathbf{1}$ denotes the all ones vectors.

This definition leads to the next lemma.

Lemma 7.9. *Let $s, t \in \mathbb{R}^m$ be any two b-separated points for some $b > 0$. Assume all coordinates of s, t are non-negative and that for every $i, j \in [m]$ there exists $S \subseteq [m]$ with $|S| = n - 1$ such that $d_{S\cup\{i\}} \neq 0$ and $d_{S\cup\{j\}\neq 0}$. Then,*

$$\phi_A\left(\frac{s+t}{2}\right) \leqslant \frac{\phi(s)+\phi(t)}{2} - b^2 \cdot \exp\left(-(n+1)(\|s\|_\infty + \|t\|_\infty)\right) \frac{\min_I d_I^2}{\det(AA^*)}.$$

We defer the proof of this lemma to Appendix A. With the previous lemma we will later argue that whenever $f(t)$ is very close to optimal, then t itself cannot be separated from an optimal solution by much.

7.3 An Algorithm

Our next theorem gives a polynomial time algorithm for computing the radial isotropic position. The assumptions are slightly stronger than simply asking that $c \in K_A$.

Theorem 7.10. *Let $\varepsilon > 0$ and $\alpha > 0$. Let $A = [u_1, \ldots, u_m] \in \mathbb{R}^{m\times n}$ with $m \geqslant n$ and $\operatorname{rank}(A) = n$. Further assume that for every $i, j \in [m]$ there exists $S \subseteq [m]$ with $|S| = n - 1$ such that $d_{S\cup\{i\}} \neq 0$ and $d_{S\cup\{j\}\neq 0}$. Then, given A and any point $c \in (1-\alpha)K_A$, we can compute a $n \times n$ matrix R such that*

$$\sum_{j=1}^m c_j \left(\frac{Ru_j}{\|Ru_j\|}\right) \otimes \left(\frac{Ru_j}{\|Ru_j\|}\right) = \mathrm{Id}_{\mathbb{R}^n} + J,$$

where $\|J\|_\infty \leqslant \varepsilon$. The running time of our algorithm is polynomial in $1/\gamma, \log(1/\varepsilon)$ and L where L is an upper bound on the bit complexity of the input A and c.

Proof: We will apply the Ellipsoid method as described in [33] (Theorem 4.1.2.) to solve the optimization problem $\sup_{t\in\mathbb{R}^m} \langle c, t\rangle - \phi_A(t)$ over the set of all $t \in \mathbb{R}^m$ satisfying $\|t\|_\infty \leqslant B$ where B is the parameter from Lemma 7.6 and $t_j \geqslant 0$ for all $j \in [m]$. Let s denote the solution computed by the Ellipsoid method and suppose we have $|f(s) - f(t^*)| \leqslant \delta^2$ where

$$\delta \leqslant \frac{\varepsilon' \cdot \min_I d_I}{e^{O(Bn)} \det(AA^*)},$$

and ε' is a sufficiently small quantity that we will bound later. With δ chosen this small it follows from Lemma 7.9 that t^* and s cannot be δ-separated. (Otherwise $\frac{s+t}{2}$ would give a solution improving the optimum.) Here, we used the fact that $\sum_{j\in I} s_j \leqslant Bn$ for every $I \subseteq [m], |I| = n$ and therefore

$$e^{\phi(s)} \leqslant e^{\log(e^{Bn}\det(AA^*))} = e^{Bn}\det(AA^*),$$

Similarly, we get the same bound for for $\phi(t^*)$. Hence, we conclude that s must be δ-close to an optimal solution in each coordinate. This implies (using standard perturbation bounds for the inverse of a matrix) that the optimality conditions from Lemma 6.1 are approximately satisfied for s in the sense that

$$e^{s_j} = \frac{(1+\varepsilon_j)c_j}{\langle u_j, (Ae^S A^*)^{-1} u_j\rangle},$$

with $\varepsilon_j \in [-\varepsilon', \varepsilon']$. Consider the positive definite matrix $M = \sum_j e^{s_j} u_j \otimes u_j$. Its inverse square root $R = M^{-1/2}$ satisfies

$$\mathrm{Id} = \sum_{j=1}^m c_j \frac{Ru_j \otimes Ru_j}{\|Ru_j\|^2} + \sum_{j=1}^m \varepsilon_j c_j \frac{Ru_j \otimes Ru_j}{\|Ru_j\|^2}.$$

Let $J = \sum_{j=1}^m \varepsilon_j c_j \frac{Ru_j \otimes Ru_j}{\|Ru_j\|^2}$ denote the error term above. It is not hard to show that for $\varepsilon' = \varepsilon/\exp(\mathrm{poly}(L)))$, we have that $\|J'\|_\infty \leqslant \varepsilon$. Since the dependence on $1/\delta$ in the Ellipsoid method is logarithmic, the running time remains polynomial in $L, 1/\alpha$ and $\log(1/\varepsilon)$. ■

Concluding Remarks

Here we gave a polynomial time deterministic estimator for d-dimensional linear regression in $\mathbb{R}^n$ that has a breakdown point of $1 - \frac{d}{n}$ and gave evidence that there is no polynomial time estimator that has a better breakdown point. We explored this question based on connections to small set expansion, matroid theory and functional analysis. The most important open question is to understand the tradeoffs between efficiency and robustness for other inference problems, for example for the more challenging problem of estimating the covariance (see [24]). Computational complexity is the obstacle to using the known robust statistical methods. Conversely it is possible that the obstacle to using known estimators with provable computational guarantees is their sensitivity to violations in the model.

Acknowledgements

Thanks to Gilad Lerman for many helpful discussions and to Joel Tropp for pointers to the literature.

References

[1] N. Alon and J. Spencer. *The Probabilistic Method*. John Wiley & Sons, 2000.

[2] F. Barthe. On a reverse form of the Brascamp-Lieb inequality. *Invent. Math.*, 134(2):335–361, 1998.

[3] H.J. Brascamp and E.H. Lieb. Best constants in Young's inequality, its converse, and its generalization to more than three functions. *Advances in Math.*, 20(2):151–173, 1976.

[4] S. Brubaker. Robust PCA and clustering in noisy mixtures. *SODA*, pages 1078–1087, 2009.

[5] E. Candes, X. Li, Y. Ma and J. Wright. Robust principal component analysis? *Journal of the ACM*, 2011.

[6] E. Candes and B. Recht. Exact matrix completion via convex optimization. *Foundations of Comp. Math.*, pages 717–772, 2009.

[7] Eric A. Carlen and Dario Cordero-Erausquin. Subadditivity of the entropy and its relation to brascamp-lieb type inequalities. *Geometric and Functional Analysis*, 19(2):373–405, 2009

[8] E. Carlen, E.H. Lieb, and M. Loss. A sharp analog of Young's inequality on S^N and related entropy inequalities. *J. Geom. Anal.*, 14(3):487–520, 2004.

[9] V. Chandrasekaran, S. Sanghavi, P. Parrilo and A. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM J. Optim.*, pages 572–596, 2011.

[10] C. Croux, P. Filzmoser and M. Oliveira. Algorithms for projection pursuit robust principal component analysis. *Chemometrics Intell. Lab. Sys.*, pages 218–225, 2007.

[11] W. Cunningham. Testing membership in matroid polyhedra. *J. Combin. Theory Ser. B*, pages 161–188, 1984.

[12] S. Dasgupta. Subspace detection: a robust statistics formulation. *COLT*, page 734, 2003.

[13] P. Davies. Aspects of robust linear regression. *Annals of Statistics*, pages 1843–1899, 1993.

[14] A. Deshpande, M. Tulsiani and N. Vishnoi. Algorithms and hardness for subspace approximation. *SODA*, pages 482–496, 2011.

[15] D. Donoho and P. Huber. The notion of breakdown point. *A Festshrift for Erich L. Lehmann*, pages 157–184, 1983.

[16] J. Dunagan and S. Vempala. Optimal outlier removal in high-dimensional spaces. *J. Comput. Syst. Sci.*, pages 335–373, 2000.

[17] H. Edelsbrunner and D. Souvaine. Computing median-of-squares regression lines and guided topological sweep. *Journal of the Amer. Stat. Assoc.*, pages 115–119, 1990.

[18] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures* (R. K. Guy et al.,), pages 69–87, Gordon & Breach, 1970.

[19] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, pages 381–395, 1981.

[20] J. Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *J. Comput. Syst. Sci.*, pages 612–625, 2002.

[21] M. Grótschel, L. Lovász and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, pages 169–197, 1981.

[22] V. Guruswami and P. Raghavendra. Hardness of solving sparse overdetermined linear systems: A 3-query PCP over integers. *ACM Trans. Comput. Theory*, pages 1–20, 2009.

[23] V. Guruswami, P. Raghavendra, R. Saket and Y. Wu. Bypassing UGC for some optimal geometric inapproximability results. *SODA*, pages 699–717, 2012.

[24] P. Huber. *Robust Statistics*. John Wiley & Sons, 1981.

[25] R. Impagliazzo, V. Kabanets. Constructive proofs of concentration bounds. *APPROX-RANDOM*, pages 617–631, 2010.

[26] S. Iwata, L. Fleisher and S. Fujishige. A combinatorial strongly polynomial time algorithm for minimizing submodular functions. *JACM*, pages 761–777, 2001.

[27] L. Khachiyan. On the complexity of approximating extremal determinants in matrices. *Journal of Complexity*, pages 138–153, 1995.

[28] S. Khot and D. Moshkovitz. Hardness of approximately solving linear equations over the reals. *STOC*, pages 413–420, 2011.

[29] P. Lax. Linear Algebra. Wiley Interscience, 2007.

[30] G. Lerman, M. McCoy, J. Tropp and T. Zhang. Robust computation of linear models, or How to find a needle in a haystack. *Arxiv*, 2012.

[31] J. Matousek. *Lectures on Discrete Geometry*. Springer-Verlag New York, Inc., 2002.

[32] A. Naor, O. Regev and T. Vidick. Efficient rounding for the noncommutative Grothendieck inequality. *Arix*, 2012.

[33] A. Nemirovski. Lectures on modern convex optimization. http://www2.isye.gatech.edu/~nemirovs/.

[34] P. Raghavendra and D. Steurer. Graph expansion and the unique games conjecture. In *STOC*, pages 755–764, 2010.

[35] P. Raghavendra, D. Steurer, and M. Tulsiani. Reductions between expansion problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:172, 2010.

[36] B. Recht, M. Fazel and P. Parrilo. Guaranteed minimum rank solutions of matrix equations via nuclear norm minimization. *SIAM Rev.*, pages 471–501, 2010.

[37] P. Rousseeuw. Least median squares regression. *Journal of the Amer. Stat. Assoc.*, pages 871–880, 1984.

[38] P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, 1987.

[39] A. Schrijver. A combinatorial algorithm for minimizing submodular functions in strongly polynomial time. *J. Combin. Theory Ser. B*, pages 346–355, 2000.

[40] S. Vempala. The joy of PCA. http://www.cc.gatech.edu/events/cse-seminar-santosh-vempala-0.

[41] H. Xu, C. Carmanis and S. Sanghavi. Robust PCA via outlier pursuit. *IEEE Trans. Inform. Theory*, pages 1–24, 2010.

[42] T. Zhang and G. Lerman. A novel M-estimator for robust PCA. *Arxiv*, 2011.

A The Defect Lemma

Here we prove Lemma 7.9:

Proof: As we did in the proof of Lemma 7.4, we will apply the Cauchy-Schwarz inequality to the vectors u, v indexed by $I \subseteq [m], |I| = n$ and defined as

$$u_I = \sqrt{e^{\sum_{j\in I} s_j} d_I}, \qquad v_I = \sqrt{e^{\sum_{j\in I} t_j} d_I}.$$

We'd like to determine how much slack we have in this inequality. Let us therefore lower bound

$$1 - \left(\frac{\langle u, v\rangle}{\|u\|\|v\|}\right)^2 = \frac{\|u\|^2\|v\|^2 - \langle u, v\rangle^2}{\|u\|^2\|v\|^2} = \frac{\frac{1}{2}\sum_{I\neq J}(u_I v_J - u_J v_I)^2}{\|u\|^2\|v\|^2},$$

where the last step is Lagrange's identity. Now write $t_j = s_j + 2a_j$. By the assumption that s, t are b-separated we must have that $\max_{i,j\in[m],i\neq j} |a_i - a_j| \geqslant b$. Let i, j be a pair of indices achieving the maximum. Without loss of generality assume that $a_j \geqslant a_i + b$. Let $S \subseteq [m]$ be a set of size $|S| = n-1$ such that $I = S \cup \{i\}$ and $J = K \cup \{j\}$ satisfy $d_I \neq 0$ and $d_J \neq 0$. Such a set S must exist by our assumption. Then:

$$\begin{aligned}(u_I v_J - u_J v_I)^2 &= \left((e^{\frac{s_i+t_j}{2}} - e^{\frac{s_j+t_i}{2}})e^{\sum_{j\in S}\frac{s_j+t_j}{2}}\right)^2 d_I d_J \\ &= \left((e^{a_j} - e^{a_i})e^{\frac{s_i+s_j}{2}} e^{\sum_{j\in S}\frac{s_j+t_j}{2}}\right)^2 d_I d_J \geqslant (e^{a_j} - e^{a_i})^2 d_I d_J\end{aligned}$$

where the inequality follows because $s_i, t_i \geqslant 0$ for all i. On the other hand, $(e^{a_j} - e^{a_i})^2 = (e^b - 1)^2 e^{2a_i}$. But $e^x - 1 \geqslant x$ and $a_j \geqslant -\|s\|_\infty$. Thus:

$$(u_I v_J - u_J v_I)^2 \geqslant \gamma \text{ with } \gamma = b^2 \min_I d_I^2 \cdot e^{-\|s\|_\infty}.$$

Therefore:

$$\left(\frac{\langle u, v\rangle}{\|u\|\|v\|}\right)^2 \leqslant 1 - \frac{\gamma}{\|u\|\|v\|}. \tag{3}$$

On the other hand

$$\|u\| = \sqrt{\det(Ae^S A^*)} \leqslant e^{n\|s\|_\infty} \det(AA^*), \qquad \|v\| = \sqrt{\det(Ae^T A^*)} \leqslant e^{n\|t\|_\infty} \det(AA^*).$$

Taking logarithms on both sides of (3), we get

$$\log\left(\frac{\langle u, v\rangle}{\|u\|\|v\|}\right) \leqslant -\frac{1}{2}\gamma \frac{e^{-n(\|s\|_\infty + \|t\|_\infty)}}{\det(AA^*)},$$

where we used that $\log(1-x) \leqslant -x$ for all $1 > x > 0$. ■